CHAPTER 15

Structuring Lines and Paragraphs with Block-Level Elements

In This Chapter

Using the P element to create paragraphs

Controlling line breaks and spacing with the BR element

Adding headings with H1 through H6

Creating indented quotations with the BLOCKQUOTE element

Adding preformatted text with the PRE element

Grouping block elements with the DIV element

You are now squarely in the middle of building your Web page. The basic building blocks of a page are block-level elements. This chapter covers them all in detail. When you finish this chapter, you will have more than 50 percent of what you need to create most Web pages. This excludes your style sheet, which is covered in Part IV.

This chapter steps up the pace a bit. If you weren't serious, you wouldn't have gotten here, so to some degree, the material starts to become more challenging. Don't worry, though, if you don't understand something. Everything glossed over in this chapter (or included in an example for purposes of demonstrating a topic) is covered in-depth later in the book. Focus on picking up the main points of this chapter; any stray items that pop up in examples are explained later.

Introducing Block-Level Elements

Block-level elements are your page's building blocks. Most of your content will go into block-level elements (unless you use the HTML 3.2 Way of putting all your content into tables, which is highly discouraged). Block-level elements serve two

purposes: They tell the browser what type of content is contained in the block and they enable you to define how that content is displayed in style sheets.

Dictating presentation

Another reason to use block-level elements is to define the layout of your page. Block-level elements are easily formatted using style sheets. Without getting into a long explanation of style sheet definitions, consider the following style sheet:

```
BODY {
        background: #FFFFFF;
        margin-left: .5in;
        margin-right: .5in;
}
BLOCKQUOTE {
        padding: .5in;
}
P {
        text-indent: .5in;
}
UL {
        margin-left: 1in;
}
LI {
        margin-left: .25in;
}
```

What does this say? It tells most of the block elements in your document how to format themselves. Why is this so great? Because it is all in one place! If you want to change your style, you can do it in a central location: in your style sheet. If you cheat and use the `BLOCKQUOTE` element for all your paragraphs, because you want a half-inch margin of white space on either side of the text, you won't have any good way to change your style — say, right-justify the entire page — without manually editing your HTML. And, if you want to add an actual blockquote to your page, you must nest `BLOCKQUOTE` elements.

Creating Paragraphs: The P Element

If your document contains prose, you'll probably want to include it in a paragraph. Do you think all documents contain text? They needn't. You can use graphics or lists effectively to communicate many messages. Creating a paragraph is easy. In the past, you might have used only a `<P>` tag at the end of each paragraph. This works, but it can produce unpredictable results with style sheets. If you plan to use style sheets, you should open and close all your paragraphs with the appropriate tags.

Paragraph <P>

Start Tag: Required

Content: Inline elements
Text

End Tag: Optional

Attributes: id, class: used in conjunction with style sheets

lang, dir: indicate language and text direction; better specified in the BODY element unless this element is an exception to the rest of the page

title: title of this element

style: define style here to override the style sheet

align: alignment is deprecated; should be specified in style sheet

events: see Chapter 48

Controlling Line Breaks and Spacing

HTML offers you two easy ways to break up your page visually: creating white space and dividing your page with lines. Many good reasons exist to use both these techniques. Generally speaking, more white space is better than less. Take advantage of white space to attract the visitor's eye.

Breaking up your page

You can use the BR element to break a line wherever you want to, even in the middle of a word. The BR element is an inline element, so it doesn't interfere with your block-level element. You can use the BR element within a block or between blocks.

Break

Start Tag: Required

Content: Empty

End Tag: Forbidden

Attributes: id, class: used in conjunction with style sheets

title: title of this element

style: define style here to override the style sheet

clear: deprecated; used to indicate where floating objects are to appear following the break; use style sheets, instead

The other way to break up your page is with horizontal lines. The HR element is just as easy to use as the BR element.

Horizontal Rule <HR>

Start Tag: Required

Content: Empty

End Tag: Forbidden

Attributes: id, class: used in conjunction with style sheets

lang: instructs the target browser to apply language rules to the text. If you entered: <HTML lang="fr"> you would be telling the browser that French rules apply.

dir: text directionality. This becomes an issue with Arabic and Hebrew because they display right to left.

title: title of this element

style: define style here to override the style sheet

align: alignment is deprecated; use style sheets, instead

noshade: deprecated; indicates the line shouldn't have a shadow, which is the default; use style sheets instead

size: deprecated; used to define the height of the rule; use style sheets instead

width: deprecated; used to define the length of the rule across the page (default is 100 percent of the width of the page); use style sheets instead

events: see Chapter 48

Preventing a line break

At times, you want to keep your line from breaking in the wrong place. You can prevent a line from breaking where you don't want it to by using a nonbreaking space, which is an entity. To add a nonbreaking space, use   as in:

```
The delegates signed the Constitution in September 1787.
```

This forces September and 1787 to appear on the same line. Be sure not to leave any spaces between September, , and 1787.

Adding Headings

Headings are another easy way to break up your page. Notice how this book takes advantage of headings. They not only make it easier to read your page, but they make it easier for visitors to find what they need when they return to your page.

There are six levels of headings: H1, H2, H3, H4, H5, and H6. H1 is the most important; H6 is the least important. Use them in order of importance, not based on how they render in your browser. That's what style sheets are for.

Heading 1 <H1> through Heading 6 <H6>

Start Tag: Required

Content: Inline elements

End Tag: Required

Attributes: id, class: used in conjunction with style sheets

lang, dir: indicate language and text direction; better specified in the BODY element unless this element is an exception to the rest of the page

title: title of this element

style: define style here to override the style sheet

align: alignment is deprecated; should be specified in style sheet

events: see Chapter 48

Creating Indented Quotations: The BLOCKQUOTE Element

The BLOCKQUOTE has been one of the most misused elements. It should be used to quote a long passage of text and, in the past, to create white space on both sides of your paragraph. Worse yet, people have used nested BLOCKQUOTE elements to get even *more* white space on their pages.

The BLOCKQUOTE element is great for quoting long passages of text. It has an attribute called cite, which, while not currently processed by browsers, could prospectively be used to link to the page where the quotation is found in its entirety.

Blockquote <BLOCKQUOTE>

Start Tag: Required

Content: Inline elements

End Tag: Required

Attributes: id, class: used in conjunction with style sheets

lang, dir: indicate language and text direction; better specified in the BODY element unless this element is an exception to the rest of the page

cite: URL of the quotation if taken from the Web

`title`: title of this element

`style`: define style here to override the style sheet

`events`: see Chapter 48

Adding Preformatted Text

Sometimes you want to show text, but you don't want the browser to have control over how it formats that text. You can force your browser to use a fixed-width font by implementing the `PRE` element. The browser also recognizes your end-of-line markers (when you hit return in your typing) to indicate the end of the line.

```
<P>The following is an example of how you use the PRE
element:</P>
<PRE>&lt;PRE&gt;Put the text you want to appear preformatted in
a PRE element.&lt;/PRE&gt;</PRE>
```

This will render as:

The following is an example of how you use the PRE element:

```
Put the text you want to appear preformatted in a PRE element.
```

Preformatted text `<PRE>`

Start Tag: Required

Content: Inline elements

End Tag: Required

Attributes: `id`, `class`: used in conjunction with style sheets

`lang`, `dir`: indicate language and text direction; better specified in the `BODY` element unless this element is an exception to the rest of the page

`width`: suggested width that the block of preformatted text should be; currently ignored by most browsers

`title`: title of this element

`style`: define style here to override the style sheet

`events`: see Chapter 48

Grouping Block Elements: The DIV Element

Just to tantalize you, you're now going to learn about the DIV element. The DIV *element* is how you group block-level elements to format them with a style sheet. Normally, you only create a style for an element, but the DIV element gives you greater flexibility by enabling you to create a zone that includes multiple block elements. Those elements can be nested, in which case inheritance would come into play, or consecutive. Consider the following example.

You have two paragraphs of text and an unnumbered list you want to draw attention to by adding a border and giving it a different background color. Without the DIV element, there is no way to put a border around multiple blocks, short of putting them all into one big cell in a table (ugly and messy). With the DIV element, you would create the following:

```
<DIV class="special">
<P>The reasons for using the DIV element are numerous:</P>
<UL>
<LI> to group block-level elements
<LI> to put a border around multiple block-level elements
<LI> to change the background color of multiple block-level
elements
</UL>
<P>Isn't HTML 4 grand?</P>
</DIV>
```

Then you would create a style sheet definition, ideally in your externally defined style sheet, to tell the browser this DIV element should have a yellow background and a blue border.

```
DIV.special {
        background: #FFFF00;
        border-top: ridge;
        border-bottom: ridge;
        border-left: ridge;
        border-right: ridge;
        padding: 15pt.;
}
```

Division <DIV>

Start Tag: Required

Content: Blocks

Inline elements

End Tag: Required

Attributes: `id`, `class`: used in conjunction with style sheets

`lang`, `dir`: indicate language and text direction; better specified in the `BODY` element unless this element is an exception to the rest of the page

`align`: deprecated; use style sheets

`title`: title of this element

`style`: define style here to override the style sheet

`events`: see Chapter 48

From Here

Jump to Chapter 26 and start learning CSS syntax.

Move on to Chapter 23 to group elements with `DIV` and `CLASS`.

Proceed to Chapter 17 and master inline elements.

Summary

You learned a lot in this chapter: how to create all the basic block-level elements, how to nest elements, and how to create three kinds of lists. You also got a taste of what the `DIV` element can do for your layout.

You learned how to break text, how not to break text, and how to insert preformatted text. You are well on your way to having mastered the essentials of HTML 4.

✦ ✦ ✦